

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Urban Peternej

# **Prototip B2B vmesnika**

DIPLOMSKO DELO

VISOKOŠOLSKI ŠTUDIJSKI PROGRAM  
PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana, 2017

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Direktna elektronska izmenjava podatkov med dvema partnerjema in njunima informacijskima sistemoma je vse bolj razširjena, saj omogoča bistveno učinkovitejše poslovanje in zmanjšuje tveganje napak. Način direktne elektronske izmenjave podatkov imenujemo B2B (business-to-business). Zasnуйте in razvijte prototip B2B vmesnika. Vmesnik naj obsega tudi enostaven pregled delovanja in izmenjave podatkov preko forme.

*Zahvaljujem se mentroju in somentorici, ker sta mi omogočila pisanje tega diplomskega dela, predvsem pa se zahvaljujem družini za vso moralno in finančno podporo v času študija.*

# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>B2B vmesnik</b>	<b>3</b>
2.1	B2B komunikacija . . . . .	3
2.2	Skica B2B vmesnika . . . . .	4
2.3	Kaj je B2B vmesnik? . . . . .	4
<b>3</b>	<b>Programska oprema</b>	<b>7</b>
3.1	Oracle JDeveloper . . . . .	7
3.2	Toad for Oracle . . . . .	7
3.3	Oracle WebLogic Server . . . . .	8
<b>4</b>	<b>Tehnologije</b>	<b>11</b>
4.1	Java . . . . .	11
4.2	Odjemalec in strežnik . . . . .	12
4.3	SOAP . . . . .	13
4.4	XML . . . . .	14
<b>5</b>	<b>Nastavitve vmesnika</b>	<b>15</b>
5.1	Pregled prenosov . . . . .	18
5.2	Integracije . . . . .	19

<b>6</b>	<b>Spletna storitev</b>	<b>21</b>
6.1	Kaj je spletna storitev . . . . .	21
6.2	Nalaganje Java servleta na Weblogic strežnik . . . . .	22
6.3	Povezava WS z grafičnim vmesnikom . . . . .	23
6.4	Metode spletne storitve . . . . .	24
6.5	SOAP sporočila . . . . .	26
6.6	Tipi datotek . . . . .	27
6.7	Nastavitve . . . . .	28
6.8	Branje in pisanje datotek v/iz baze . . . . .	30
6.9	Pisanje logov . . . . .	31
6.10	Shranjevanje v arhiv . . . . .	32
6.11	Načini klicanja . . . . .	33
6.12	Od lokalne do produkcijske verzije . . . . .	36
<b>7</b>	<b>Sklepne ugotovitve</b>	<b>39</b>
	<b>Literatura</b>	<b>41</b>

# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>B2B</b>	business to business	medpodjetniško
<b>WS</b>	web service	spletna stritev
<b>XML</b>	extensible markup language	razširljivi označevalni jezik
<b>SOAP</b>	Simple Object Access Protocol	protokol za izmenjavo podatkov med spletnimi storitvami
<b>WSDL</b>	Web Services Description Language	protokol za definicijo spletnih storitev





# Povzetek

**Naslov:** Prototip B2B vmesnika

**Avtor:** Urban Peternej

B2B je v našem primeru način elektronske izmenjave podatkov, ki omogoča varno in zanesljivo izmenjavo sporočil med računalniškima rešitvama partnerja in banke. Nastal je zaradi potrebe po izmenjavi sporočil plačilnega prometa med udeleženima stranema.

B2B vmesnik je sestavljen iz grafičnega vmesnika ter iz spletnega servisa, ki teče v ozadju. Vmesnik skrbi za grafični prikaz prenosov v obe strani, nastavitve in integracije, spletni servis pa za vse operacije, ki se izvajajo v ozadju. S pomočjo spletnega servisa tako odlagamo datoteke na »outbox«, jih sprejemamo iz »inboxa«, pišemo v bazo in iz baze, prenašamo datoteke v arhiv in pišemo uspešne ter neuspešne loge v bazo. Spletni servis nam tako olajša izmenjavo datotek s partnerji ter njihovo shranjevanje.

**Ključne besede:** B2B, vmesnik, Java, Oracle.



# Abstract

**Title:** Prototype of B2B interface

**Author:** Urban Peternelj

B2B is a type of electronic exchange of messages between partner's and bank's solutions. It makes exchange of messages safe and reliable. This communication was created for the exchange of payment transactions messages.

B2B interface consists of a GUI and a Web service that runs in the background. The interface provides a graphical representation of file transfer in both sides, settings and integration, meanwhile the web service executes operations carried out in the background. By using the online service we drop the files on the "outbox", we take them from the "inbox", write files in the archive and write successful and unsuccessful logs to base. The web service allows us to exchange the files with partners and store them.

**Keywords:** B2B, inteface, Java, Oracle.



# Poglavje 1

## Uvod

B2B vmesnik bistveno poenostavi in pospeši izmenjavo sporočil med banko in partnerji. Poleg tega so v komunikacijski vmesnik vgrajeni preprosti, a trdoživi mehanizmi, ki zagotavljajo zanesljiv prenos podatkov. Vgrajeni so napredni varovalni elementi, ki temeljijo na asimetrični kriptografiji in infrastrukturi PKI in ki zagotavljajo varen prenos podatkov po javnem omrežju.

B2B vmesnik tako izboljšuje dozdajšnji način izmenjave datotek in podatkov, rezultati pa bodo koristili tako banki, njenim uslužbencem kot tudi odjemalcem, torej partnerjem.

Rezultat diplomskega dela bo delujoči B2B vmesnik s podporo spletnega servisa. Vmesnik bo uporabljen v vsakodnevni prenosih podatkov med ponudnikom in odjemalci. Končna rešitev bo kombinacija že obstoječega grafičnega vmesnika in na novo postavljenega spletnega servisa, ki bo dopolnjeval obstoječi vmesnik. Končno rešitev bodo preverili banka in njeni uslužbenci ob vsakodnevni uporabi B2B vmesnika.



# Poglavje 2

## B2B vmesnik

### 2.1 B2B komunikacija

Business-to-business (B2B) predstavlja komunikacijo, pri kateri se en poslovni partner poveže z drugim. To se običajno zgodi, ko:

- Poslovni partner pridobiva materiale za proizvodni proces (npr. proizvajalec hrane kupi sol);
- Poslovni partner potrebuje storitve drugega partnerja (npr. proizvajalec hrane zaposluje računovodsko podjetje za revidiranje svojih financ);
- Poslovni partner prodaja blago in storitve, ki jih proizvaja nekdo drug (npr. trgovec nakupuje končni izdelek od proizvajalca hrane).

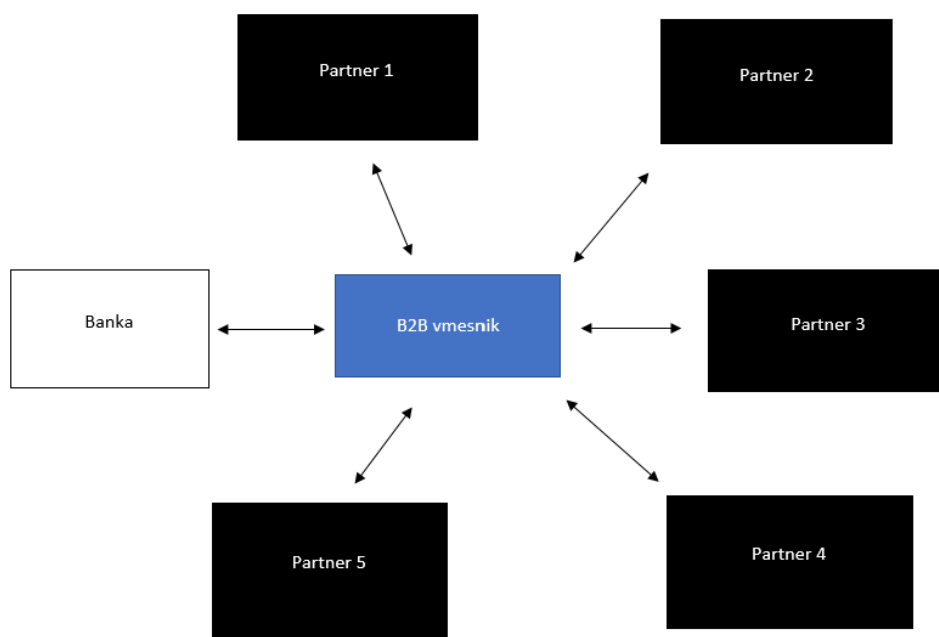
B2B je pogosto nasprotna vrsta komunikacije, kot jo poznamo med podjetji in potrošniki (B2C). V B2B poslovanju se pogosto zgodi, da imajo stranke v razmerju primerljivo pogajalsko moč, in tudi če je nimajo, vsaka stran praviloma vključuje strokovno osebje in pravnega svetovalca pri pogajanjih o pogojih, medtem ko je B2C oblikovan tako, da ima veliko večjo moč gospodarska stran.

V primeru banke in njenih partnerjev komunikacija poteka preko B2B vmesnika, kjer ena izmed strani v vmesnik odloži zeleno datoteko, nato pa jo B2B

avtomatsko posreduje želenemu prejemniku. Druga stran datoteko pridobi iz vmesnika in po potrebi v vmesnik kot odgovor odloži novo datoteko, ki bo posredovana nazaj pošiljatelju.

## 2.2 Skica B2B vmesnika

Na spodnji sliki je prikazana vloga B2B vmesnika med banko in partnerji ter smeri prenosa datotek med banko in partnerji.



Slika 1: Skica delovanja B2B vmesnika

## 2.3 Kaj je B2B vmesnik?

B2B vmesnik je javen in odprt komunikacijski vmesnik, ki omogoča varen in zanesljiv prenos podatkov med odjemalcem in ponudnikom. B2B vme-



---

snik je sicer nastal z namenom opredelitve standardne izmenjave sporočil, ki so usklajena s podatkovnim standardom XML, vendar pa so vmesniki, protokol in načela izmenjave podatkov kljub temu dovolj splošni, da jih lahko uporabimo za prenos poljubnih podatkov. [9]



## Poglavje 3

# Programska oprema

Za načrtovanje in razvoj smo uporabljali programske rešitve proizvajalca Oracle, ki so najbolj primerne za javanski razvoj in povezavo le-teh na podatkovno bazo.

### 3.1 Oracle JDeveloper

JDeveloper je brezplačna programska oprema, ki jo je razvilo podjetje Oracle Corporation, njeni začetki pa segajo v leto 1997. Oracle JDeveloper je integrirano razvojno okolje (IDE) za programiranje v Javi, ponuja pa tudi okolje za razvoj v programskih jezikih, kot so: XML, SQL in PL / SQL, HTML, JavaScript, BPEL in PHP. Ponuja široko paleto možnosti, kot so oblikovanje, razvoj, organizacija kode in nalaganje javanskih programov na strežnik. JDeveloper je razvojno okolje, ki vsebuje veliko orodij in generatorjev kode, ki olajšajo implementacijo javanskih rešitev in uporabniku omogoči, da se osredotoči na razvoj poslovnih rešitev. [4]

### 3.2 Toad for Oracle

Jim McDaniel ("TOADman") je prvotno orodje razvil za lastno uporabo v poznih devetdesetih. Imenoval jo je "Tool for Oracle Application Develo-

pers”, na kratko TOAD. McDaniel je orodje na začetku na svetovnem spletu objavil kot brezplačno programsko opremo. [5]

Toad Software je danes orodje za upravljanje baz podatkov proizvajalca Quest, ki ga razvijalci baz podatkov, skrbniki baz podatkov in podatkovni analitiki uporabljajo za upravljanje tako relacijskih kot nerelacijskih baz podatkov z uporabo SQL-a. Uporablja proaktivni pristop k upravljanju podatkov in se osredotoča na današnje gospodarstvo, ki temelji na podatkih.

Rešitve v Toadu omogočajo avtomatizacijo procesov, zmanjšajo tveganja in časovno obdobje za razvoj projektov zmanjšajo za skoraj polovico.

Toad omogoča tudi naslednje funkcije:

- izvaja dosledne in ponovljive procese, ki podpirajo agilni razvoj podatkovnih baz;
- pospeši razvoj aplikacij, hkrati pa zmanjša tveganja, povezana s spremembami baze podatkov;
- zagotavlja funkcionalno natančnost in razširljivost s samodejnim testiranjem;
- hitro določi in odpravi neučinkovitosti delovanja baze podatkov;
- avtomatizira optimizacijo SQL;
- avtomatizira in načrtuje zapletene ali rutinske naloge baze podatkov.

[8]

### 3.3 Oracle WebLogic Server

Oracle WebLogic Server je prva svetovna oblačna Java platforma, ki nam omogoča, da v celoti izkoristimo prednosti računalništva v oblaku. Lahka razdrobljena arhitektura poganja izolirane aplikacije in zagotavlja popolno prenosljivost med zasebnim in javnim oblakom. Vse skozi razpoložljiv center ščiti aplikacije pred motnjami. Poleg tega WebLogic omogoča tudi dostop do

platforme kjer koli s pomočjo orodja Oracle Java Cloud Service, ki temelji na Oracle WebLogic Serverju.

Oracle WebLogic temelji na javanski platformi in omogoča izdelavo in objavo javanskih programov na spletnem strežniku. Administratorska konzola omogoča lažje nastavljanje strežnika, poleg tega pa vsebuje tudi podporo SSL podatkovne enkripcije ter avtentikacijske in avtorizacijske mehanizme. [2]



# Poglavje 4

## Tehnologije

### 4.1 Java

Java je objektno usmerjeni, prenosljivi programski jezik, ki ga je razvil James Gosling s sodelavci. Projekt, ki se je na začetku razvoja leta 1991 imenoval Oak, je bil razvit v podjetju Sun Microsystems kot zamenjava za C++. Različica Java 1.0 je bila objavljena leta 1996, zadnja različica, verzija 8.0, pa je izšla marca 2014. Java vzdržuje in posodablja podjetje Oracle - Sun Microsystems.

Java je objektno usmerjena. Objektno usmerjeni jeziki so najnovejši in najuspešnejši pristopi k programiranju. Objektno programiranje je postalo priljubljeno ob koncu devetdesetih in je ostalo tudi v novem tisočletju. Java je popolnoma objektno usmerjena že od nastanka in ji objekti niso bili dodani kasneje, kot pri nekaterih drugih jezikih.

Java podpira internet. Glavna motivacija pri uporabi Java je, da omogoča razvoj programov, ki podpirajo uporabo na internetu. Java programi so lahko brez težav klicani iz spletnih brskalnikov, kot so Google Chrome, Mozilla Firefox ... Poleg tega se lahko javanski programi brez težav prenašajo preko interneta in tečejo na katerem koli računalniku.

Java je večnamenska. Čeprav je bila Java ustvarjena za pisanje spletnih aplikacij, je vseeno večnamenski programski jezik. Vse, kar lahko počnejo

C++, javaScript in ostali programski jeziki, lahko počne tudi Java.

Java je neodvisna od platforme. Java lahko teče nespremenjena skoraj na vsakem računalniku, kar omogočajo le redki programski jeziki, zato Java tudi uporablja slogan "write once - run anywhere".

Java je robustna. Če se javanski program pokvari oziroma je v njem napaka, ne bo povzročila hujše škode ali sesutja sistema. Java namreč teče v "kletki", zato so vse napake obvladljive.

Dodatne knjižnice. Ker je Java majhen programski jezik, večino dodatnih funkcionalnosti dobimo z dodatnimi knjižnicami. Celotna paleta dodanih knjižnic nam omogoča, da prikazujemo grafiko, dostopamo do interneta in izdelujemo uporabniške grafične vmesnike.[1]

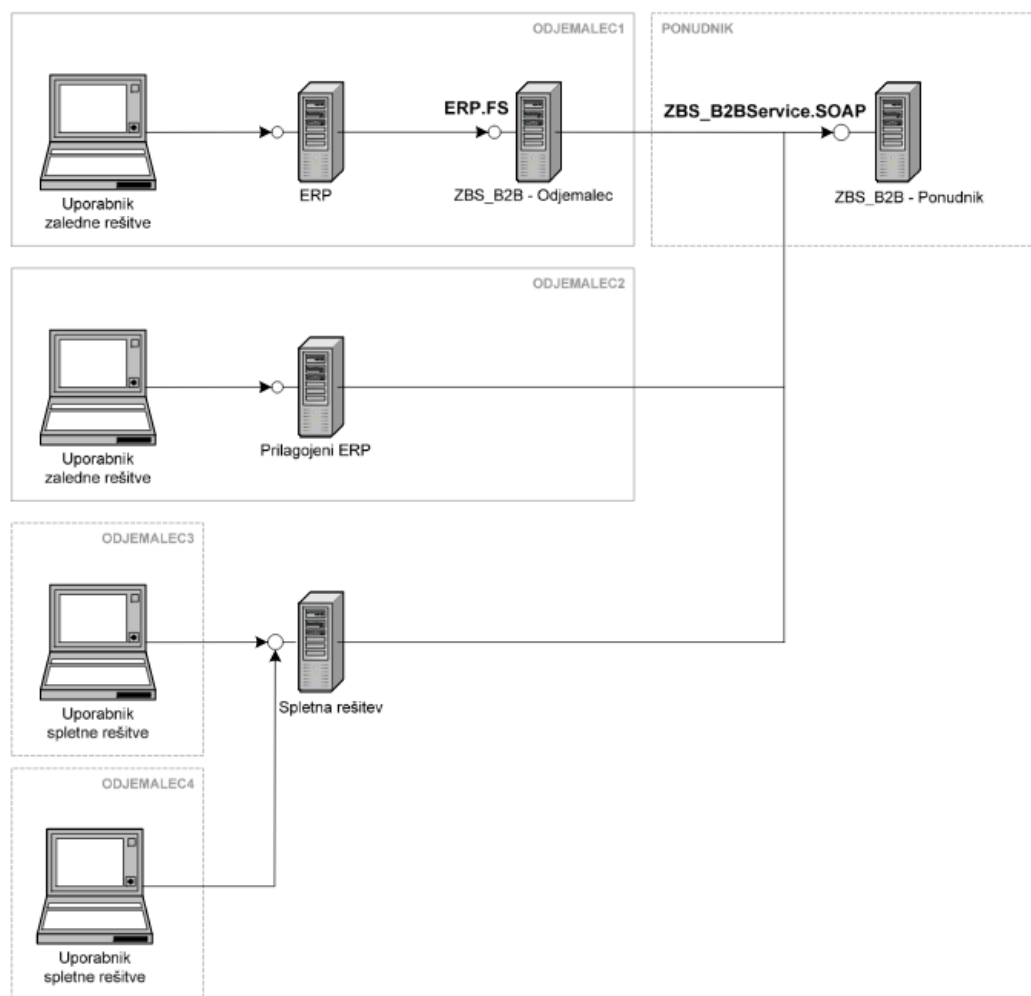
## 4.2 Odjemalec in strežnik

Spodnja slika prikazuje tipično arhitekturo sistema, ki omogoča odjemalcu izmenjavo sporočil s ponudnikom. Odjemalec uporablja npr. zaledno programsko rešitev, zato se za končnega uporabnika te rešitve izmenjava sporočil ne razlikuje od drugih, vsakdanjih zalednih opravil. Zaledna rešitev se na ponudnika povezuje posredno, s pomočjo komunikacijskega vmesnika B2B - Odjemalec, nameščenega v odjemalčevem okolju. Komunikacijski vmesnik je rešitev, ki zaledni rešitvi izpostavi prijazen in preprost vmesnik, skrije zapletene varovalne elemente in poskrbi za druge izvedbene podrobnosti komunikacijskega protokola B2B.

Standardni vmesnik B2B je pri ponudniku izpostavljen skozi rešitev "B2B - Ponudnik", ki na ponudnikovi strani poskrbi za varovalne elemente in druge izvedbene podrobnosti komunikacijskega protokola B2B. Integracija rešitve v zaledni ponudnikov informacijski sistem "Ponudnikov IS" omogoča vmesniku dejansko izmenjevanje sporočil. B2B predpisuje način izmenjave sporočil med odjemalcem in ponudnikom. Na sliki je prikazan kot komunikacija, ki jo izpostavljata spletni storitvi B2BService.SOAP pri ponudniku in B2BClient.SOAP (ki ni obvezni del standarda) pri odjemalcu. Slika spodaj prikazuje



le primer arhitekture rešitve. Standard B2B nikoli ne zahteva takšne arhitekture. Odjemalčeva ali ponudnikova rešitev sta lahko zasnovani tudi na drugačen način.



Slika 2: Arhitektura odjemalec - strežnik [9]

### 4.3 SOAP

Komunikacijski protokol standarda B2B temelji na protokolu SOAP. SOAP je standardni protokol, ki s pomočjo sporočil XML omogoča izmenjavo podatkov po računalniških omrežjih v porazdeljenem okolju. Izmenjava sporočil je izvedena skozi dve spletni storitvi. B2BService je spletna storitev pri ponu-

dniku, spletna storitev B2BClient pri odjemalcu je neobvezni del standarda. Namenjena je naprednejšim odjemalcem, ki jim omogoča bolj učinkovito izmenjavo sporočil s ponudnikom.

## 4.4 XML

Beseda XML je okrajšava za angleški izraz Extensible Markup Language, kar pomeni razširljivi označevalni jezik. XML prek značk (ang. tags), ki jih uporabnik ustvari sam in niso vnaprej določene, določa pravila za sestavljanje dokumentov v formatu, ki je razumljiv tako človeku kot stroju. XML spreminja mnogo področij računalništva, še posebej področje komunikacije med aplikacijami in strežniki, saj le-to zelo poenostavi. XML pa lahko tudi razširimo (zato tudi ime razširljivi označevalni jezik), saj ima možnost, da lahko sami določimo imena značk. Zelo je uporaben za komunikacije, saj je zgrajen zelo preprosto in pregledno. XML je razdeljen na 3 dele:

- podatkovni (v ta del shranimo poljubne podatke v neki obliki s pomočjo željenih značk),
- deklarativni (skrbi za to, da lahko pri dodajanju novih podatkov vidimo, kaj določena značka predstavlja),
- predstavitevni (z njim oblikujemo izpis podatkov). [7]

## Poglavje 5

# Nastavitve vmesnika

Z Windows odjemalcem, torej B2B Nadzorno aplikacijo, ki je uporabniku dostopna iz lokalnega računalnika preko namizne aplikacije, lahko upravljamo z nastavitvami sistema B2B. V nadzorni aplikaciji lahko nastavljamo in pregledujemo naslednje podatke:

- Splošne nastavitve
  - Digitalno potrdilo odjemalca: digitalno potrdilo odjemalca SSL, s katerim se naše podjetje ali banka predstavi ostalim odjemalcem. Nastavitev je vidna samo v primeru, če smo v komunikacijo udeleženi kot ponudnik. Prikazan je tudi datum veljavnosti digitalnega potrdila. Če je obarvan v rdečo barvo, pomeni, da nam je digitalno potrdilo preteklo.
  - Perioda izvajanja prenosa podatkov: določa časovni interval za izvedbo prenosa podatkov. S pritiskom na gumb "Prenesi takoj" podatke prenesemo takoj.
  - Velikost delov paketa: največja možna velikost posameznega poslanega dela paketa.
  - Briši podatke starejše od: po izteku intervala, ki smo ga nastavili

za brisanje zgodovine podatkov, se izbrišejo dnevniške datoteke, vsebine map C: \InboxArchive, C: \OutboxError, C: \InboxArchive in zgodovina prenosov, ki jih lahko vidimo v uporabniškem vmesniku pod zavihkom »Pregled prenosov«.

- Največja velikost datotek: določimo največjo možno velikost datoteke, ki je lahko oddana ali sprejeta iz B2B vmesnika.
- Časovna omejitev: najdaljši možni čas čakanja na odziv po tem, ko eden izmed partnerjev kliče spletno storitev.
- Maksimalni čas dostave: vpišemo največje število ur, ko poizkušamo paket dostaviti partnerju. V kolikor pošiljanje ne uspe, ga premaknemo v mapo »OutboxError«.
- Informativna spletna stran: naslov spletne strani (če jo imamo), ki vsebuje informacije o naši vlogi v sistemu B2B ter načinu obdelave paketov.
- Urnik prenosov: časovni intervali, v katerih se izvaja prenos podatkov. Nastavitev urnika ni obvezna in če urnika nismo nastavili, se prenos izvaja ves čas.

- Ponudniki

Prikazuje nastavitve podatkov o ponudnikih. Pri dodajanju ali urejanju podatkov o ponudniku je potrebno nastaviti naslednje podatke:

- aktivacija ponudnika,
- ID ponudnika,
- ime ponudnika,
- naše digitalno potrdilo odjemalca SSL, s katerim se naše podjetje oziroma banka predstavi ponudniku,
- digitalno potrdilo odjemalca SSL, s katerim se ponudnik predstavi nam; te nastavitve ne uporabljamo, če smo običajni odjemalec (če je datum veljavnosti digitalnega potrdila že pretekel, se obarva rdeče),

- Urnik prenosov: časovni intervali, v katerih ponudnik obdeluje prejete pakete. Če urnik ni nastavljen, ponudnik obdeluje pakete ves čas.

- Odjemalci

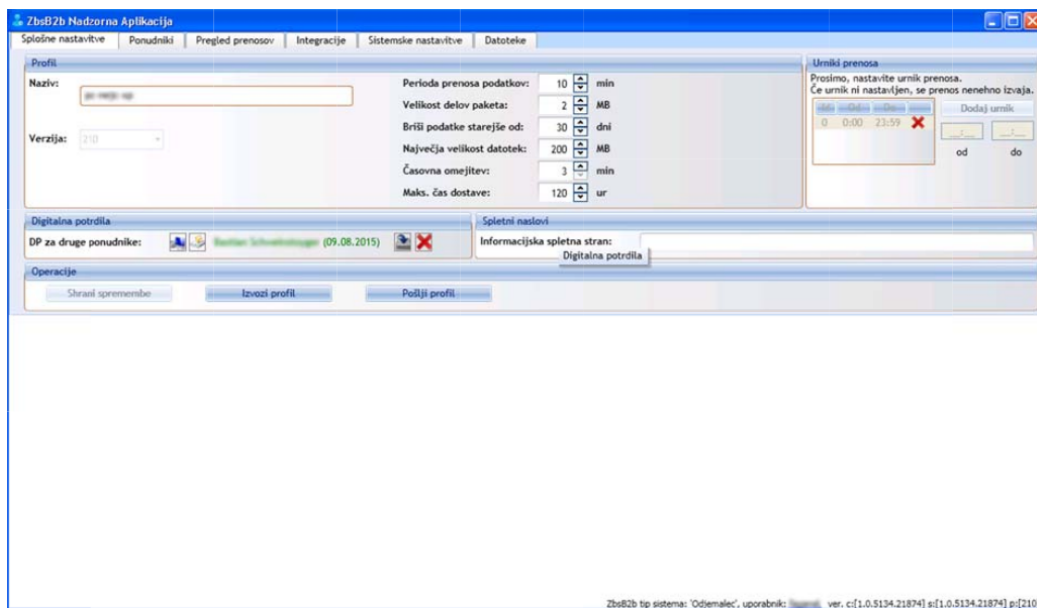
Prikazuje nastavitve podatkov o odjemalcih. Nastavitve so vidne samo, če se predstavljamo kot ponudnik. Pri urejanju ali dodajanju podatkov o odjemalcu moramo nastaviti naslednje podatke:

- aktivnost odjemalca,
- ID odjemalca,
- ime odjemalca,
- digitalno potrdilo odjemalca SSL, s katerim se nam odjemalec predstavi (če je datum prikazan z rdečo barvo, pomeni, da je digitalno potrdilo že preteklo).

- Integracija

Nastavitve povezav z zalednimi rešitvami in opredelitev, ali se prenos omejuje samo na v tabeli določene tipe datotek. Za vsak tip datoteke določimo naslednje podatke: tip integracije (WS – pokliče se spletna storitev, EXE – pokliče se izvršni program .exe, FS – datoteka se le prenese na datotečni sistem, brez nadaljnje integracije), tip datoteke, URL spletne storitve oz. pot do izvršne datoteke .exe in konstantni parametri, ki se prenesejo ob klicu spletne storitve oz. izvršne datoteke .exe. V primeru, da možnost »Prenos se ne omejuje na izbrane tipe datotek« ni izbrana, se bodo prenesle datoteke, katerih tip je naveden v polju »Tip Datoteke«. V primeru, da je možnost »Prenos se ne omejuje na izbrane tipe datotek« izbrana, se bodo prenesle vse datoteke, ne glede na tip.

- Datoteke pregled map za komunikacijo v B2B vmesniku, ki vključuje tudi možnost »Prenesi takoj«.



Slika 3: Nastavitve B2B vmesnika

## 5.1 Pregled prenosov

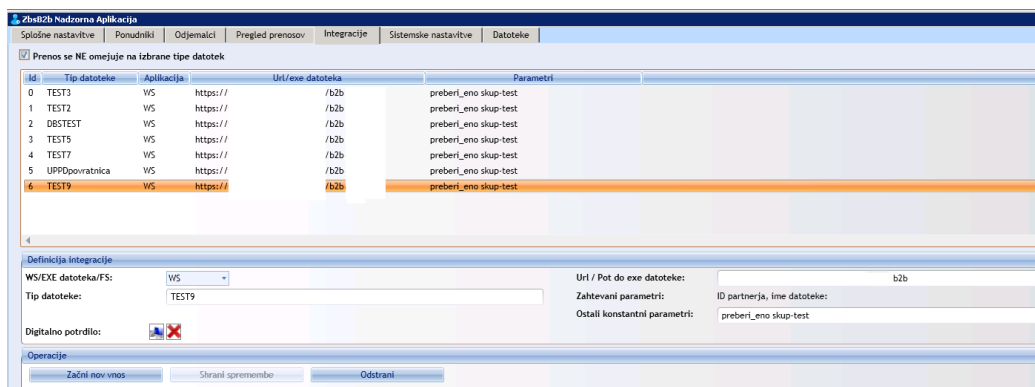
V zavihku "Pregled prenosov" lahko nadzorujemo vse prejete in oddane podatke ter njihove statuse. Za vsak prenos lahko preverimo zaporedno številko, datum prenosa, naziv ponudnika ter odjemalca, tip prenosa in datoteke, naziv datoteke in po možnosti opis napake. Vsak prenos je obarvan zeleno ali rdeče, kar kaže na uspešnost oz. neuspešnost prenosa. V primeru uspešnega prenosa bo vrstica obarvana zeleno, v primeru neuspešnega prenosa pa se bo vrstica obarvala rdeče, napaka pa bo izpisana v stolpcu opis napake. Če pride do napake, podatki niso bili uspešno preneseni in jih moramo poslati ponovno.

Splošne nastavitve								
Ponudniki			Odjemalci		Pregled prenosov		Integracije	
Sistemske nastavitve			Datoteke					
Iskalnik								
Datum in čas prejema od:			23. marec 2017 0:00:00		do		15. april 2017 0:00:00	
Tip datoteke:								
Naziv ponudnika:			Prikaži		<input type="radio"/> vse		<input type="radio"/> uspešne	
Naziv odjemalca:					<input type="radio"/> z napako		<input type="button" value="Prikaži"/>	
Id	Datum	Naziv ponudnika	Naziv odjemalca	Tip prenosa	Tip datoteke	Naziv datoteke	Št. neus	Opis napake
2638	12.04.2017 07:32:50		DBSRAZVOJ	prejeto od odjemalca	TEST7	DBSRAZVOJ_TEST7_20170407.XML	0	
2637	12.04.2017 07:15:48		DBSRAZVOJ	prejeto od odjemalca	TEST7	DBSRAZVOJ_TEST7_20170407.XML	0	
2636	12.04.2017 07:13:56		DBSRAZVOJ	prejeto od odjemalca	TEST5	DBSRAZVOJ_TESTS_20141027.XML	0	
2635	12.04.2017 07:12:06		DBSRAZVOJ	prejeto od odjemalca	TEST3	DBSRAZVOJ_TEST3_XXXX.TXT	0	
2634	12.04.2017 07:12:05		DBSRAZVOJ	prejeto od odjemalca	TEST2	DBSRAZVOJ_TEST2_25012017.txt	0	
2633	12.04.2017 07:12:00		DBSRAZVOJ	prejeto od odjemalca	DBSTEST	DBSRAZVOJ_DBSTEST_TEST2.TXT	0	
2632	11.04.2017 15:22:23	UPPD		prejeto od ponudnika	UPPDpovratnica	UPPD_UPPDpovratnica_UPPDpovratnica2017041	0	
2631	11.04.2017 15:17:23	UPPD		poslano ponudniku	DBSP00	UPPD_DBSP00_GOT201700012.xml.gpg	0	
2630	11.04.2017 14:52:21	UPPD		prejeto od ponudnika	UPPDpovratnica	UPPD_UPPDpovratnica_UPPDpovratnica2017041	0	
2629	11.04.2017 14:47:21	UPPD		poslano ponudniku	DBSP00	UPPD_DBSP00_NAW201700002.xml.gpg	0	
2628	11.04.2017 14:36:16		DBSRAZVOJ	poslano odjemalcu	TEST5	DBSRAZVOJ_TESTS_20141027.XML	0	
2627	11.04.2017 14:27:20	UPPD		prejeto od ponudnika	UPPDpovratnica	UPPD_UPPDpovratnica_UPPDpovratnica2017041	0	
2626	11.04.2017 14:22:19	UPPD		poslano ponudniku	DBSP00	UPPD_DBSP00_NAW201700004.xml.gpg	0	
2625	11.04.2017 14:07:18	UPPD		prejeto od ponudnika	UPPDpovratnica	UPPD_UPPDpovratnica_UPPDpovratnica2017041	0	
2624	11.04.2017 14:02:17	UPPD		poslano ponudniku	DBSP00	UPPD_DBSP00_GOT201700005.xml.gpg	0	
2623	11.04.2017 13:40:27		DBSRAZVOJ	prejeto od odjemalca	TEST7	DBSRAZVOJ_TEST7_20170407.XML	0	
4								
Podrobni opis								
Id:		2732			Naziv ponudnika:			
Datum:		13.04.2017 07:38:15			Naziv odjemalca:		DBSRAZVOJ	
Tip prenosa:		prejeto od odjemalca			Naziv datoteke:		DBSRAZVOJ_TEST7_20170407.XML	
Tip datoteke:		TEST7			Št. neusp. klicev:		0	
Opis napake:								
Operacije								
Prenesi takoj			Izvozi . csv					

Slika 4: Pregled prenosov

## 5.2 Integracije

Pri prejemanju in oddajanju podatkov moramo ločiti partnerje po enoličnih imenih in za vsakega nastaviti primerne nastavitve. V zavihku "Integracije" tako nastavimo klic WS, tip datoteke (ime partnerja), spletni naslov našega WS ter dodatne parametre, ki jih bomo potrebovali pri dodatni obdelavi podatkov. Ime partnerja in pot do datoteke bosta ob klicu WS podana avtomatsko. V primeru, da dobimo novega partnerja, ga le dodamo na seznam integracij.



Slika 5: Pregled integracij



# Poglavje 6

## Spletna storitev

### 6.1 Kaj je spletna storitev

Izraz spletna storitev opisuje standardiziran način vključevanja spletnih aplikacij s pomočjo XML, SOAP, WSDL in UDDI standardov prek internetnih protokolov. XML je format podatkov, ki se uporablja za zapis in prenos podatkov in zagotavlja metapodatke, povezane z datoteko, WSDL se uporablja za opis spletne storitve in UDDI navede, katere storitve so na voljo.

Mnoge organizacije uporabljajo različne sisteme programske opreme za upravljanje. Različni sistemi programske opreme pogosto za izmenjavo podatkov med seboj zahtevajo nekakšen vmesnik in spletna storitev je način komunikacije, ki omogoča, da dva različna programska sistema izmenjujeta podatke preko interneta. Sistem programske opreme, ki zahteva podatke, se imenuje "prosilec storitve", sistem programske opreme, ki obravnava zahtevek in predloži odgovor pa se imenuje "ponudnik storitve".

Različna programska oprema lahko uporablja različne programske jezike, zato je potreben način izmenjave podatkov, ki ni odvisen od določenega programskega jezika. Večina vrst programske opreme podpira tip datoteke XML. Tako lahko spletne storitve uporabljajo datoteke XML za izmenjavo podatkov.

Pravila, ki jih je potrebno potrebno opredeliti za komunikacijo med različnimi

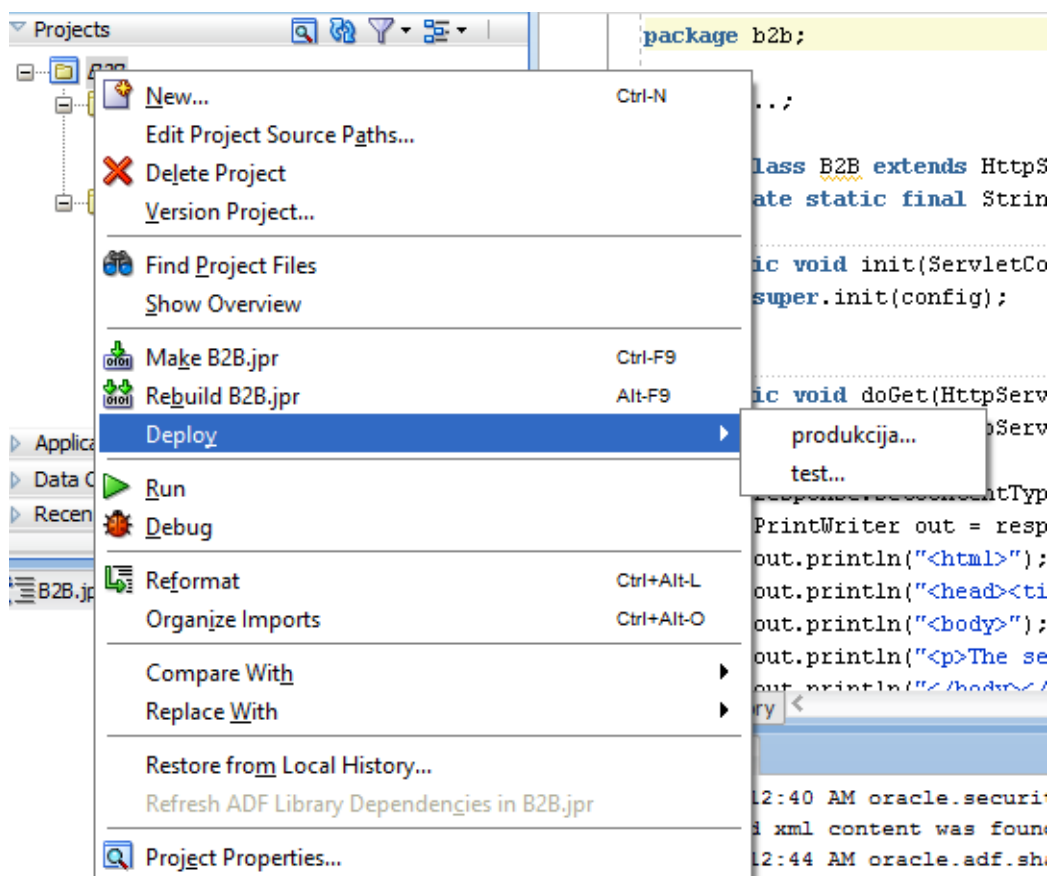
sistemi, so: kako lahko en sistem zahteva podatke iz drugega sistema, kateri posebni parametri so potrebni za zahtevo podatkov in kakšna naj bi bila struktura danih podatkov. [6]

## 6.2 Nalaganje Java servleta na Weblogic strežnik

Če želimo, da bo naša spletna storitev dosegljiva uporabnikom in B2B vmesniku jo moramo naložiti na Oraclov Weblogic strežnik. To najlažje storimo z orodjem, kjer je bila spletna storitev tudi napisana, torej z jDeveloperjem, ki ima Weblogic strežnik že integriran. Tako lahko na strežnik aplikacije nalagamo in jih tudi testiramo, preden jih naložimo na strežnik, kjer bodo dostopne tudi ostalim uporabnikom.

Ko smo s svojo aplikacijo zadovoljni jo naložimo na javno dostopen strežnik. v jDeveloperju ustvarimo nov profil za nalaganje in aplikacijo nato prek novo ustvarjenega profila naložimo na željeni strežnik. Pri raznih nadgradnjah servleta ga bomo tako lahko vedno na novo naložili preko že ustvarjenega profila.

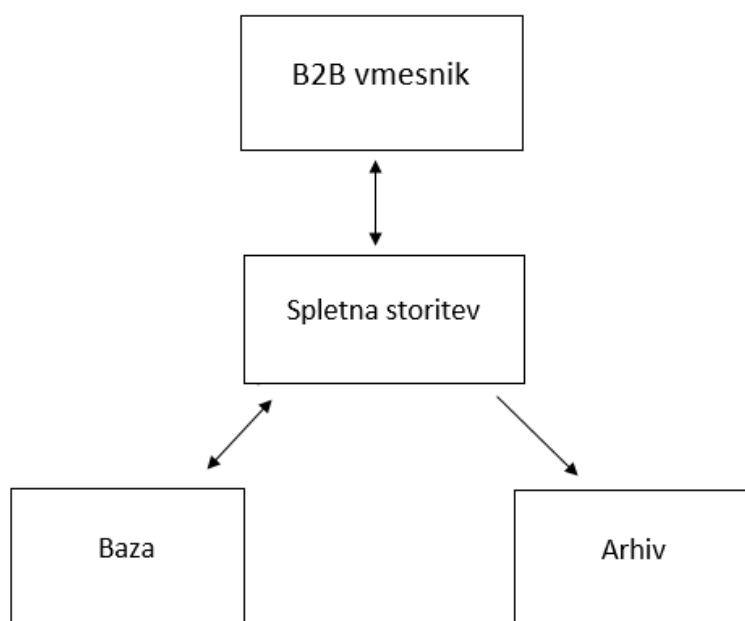
Z več ustvarjenimi profili tako lahko kasneje ob kakršnihkoli spremembah novo rešitev najprej zaženemo in testiramo lokalno, nato jo naložimo na testni strežnik in šele kasneje na produkcijski.



Slika 6: Če imamo nastavljenih več profilov, lahko izbiramo na kateri strežnik bomo naložili aplikacijo

### 6.3 Povezava WS z grafičnim vmesnikom

Povezavo grafičnega vmesnika na spletni servis določimo v nastavitvah vmesnika in v zavihku "Integracije". WS deluje kot posrednik med grafičnim vmesnikom na eni strani in bazo ter arhivom na drugi. Vsi prenosi in operacije, ki se izvedejo, potujejo preko spletne storitve.



Slika 7: Vloga spletne storitve

## 6.4 Metode spletne storitve

- Preberi eno

Metoda se avtomatsko izvede ob prihodu vsake nove datoteke v B2B vmesnik. Poskrbi, da se datoteka prenese na željeno lokacijo (v bazo ali direktorij) ter v arhivski direktorij vseh sprejetih datotek. Metode "preberi eno" ne bomo implementirali v nobeno izmed dodatnih aplikacij za klic spletne storitve, ker se bo v vseh primerih izvedla avtomatsko in zato dodatni klici ne bodo potrebni oziroma niti ne bi bili učinkoviti. Parametri, ki bodo podani ob klicu metode preberi eno:

- type - tip metode (v tem primeru "preberi eno"),
- id partnerja – partner, ki nam je poslal datoteko na B2B vmesnik,
- ime datoteke,

- ds - data source za prijava na bazo.

Slika 8: Primer klica metode preberi eno v B2B vmesniku

- Preberi

Metoda prenese vse datoteke iz direktorija, ki ga podamo preko enega izmed argumentov v bazo ali na določen direktorij ter vedno tudi v arhiv. Če ima podani direktorij tudi poddirektorije, iz njih prav tako prebere datoteke in jih shrani na zelene lokacije. Po prenosu metoda vse datoteke tudi izbriše. Parametri metode:

- type - tip metode (v tem primeru "preberi"),
- readdir - direktorij od koder preberemo podatke. Na primer: C:/B2B/INBOX/TEST3,
- ds - data source za prijavo na bazo.

- Pošlji

Klicu te metode podamo identifikacijsko številko oziroma številko sekvence datoteke iz baze, nato pa metoda poišče pripadajočo datoteko in jo prenese na outbox direktorij. V kolikor podamo napačno identifikacijsko številko, na primer, da le-ta v bazi ne obstaja, nam program javi napako. Po prenosu na B2B datoteko kopira še v arhivski direktorij za oddajo ter popravi status in datum prenosa v tabeli, kjer je bila datoteka shranjena. Parametri metode:

- PartnerId – ime partnerja kateremu je datoteka namenjena,
- PackageFileName - neuporabljen parameter,
- otherParameters:
  - \* type - tip metode (v tem primeru "pošlji"),
  - \* sekvenca datoteke - zaporedna številka datoteke, ki jo želimo odložiti na "outbox",
  - \* ds - ime data source-a za dostop do baze.

## 6.5 SOAP sporočila

Kot smo že omenili, komunikacija med B2B vmesnikom in spletno storitvijo poteka preko XML SOAP ovojnic. Glavne značke, ki jih bomo pri tem uporabljali so, "partnerID", "packageFileName" in "otherParameters" (uporaba parametrov za posamezne metode je opisana v prejšnji točki). V značko "partnerID", kot že ime pove, zapišemo ime partnerja, značko "packageFileName" lahko načeloma pustimo tudi prazno, saj bomo pot do direktorija podali v naslednji znački. V znački "otherParameters" pa podamo ime ene izmed metod "pošlji" ali "preberi", pot do direktorija in ime baze.

```
<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Body>
    <ProcessData>
      <ProcessDataRequest>
        <partnerId>TEST</partnerId>
        <packageFileName>C:\B2b\B2bServis\Inbox\TEST_XXXX4.TXT</packageFileName>
        <otherParameters>preberi C:\B2B\INBOX baza-test</otherParameters>
      </ProcessDataRequest>
    </ProcessData>
  </soap:Body>
</soap:Envelope>
```

Slika 9: Primer zahtevka v XML obliki

Po poslani zahtevi dobimo tudi SOAP odgovor. Glavni znački, ki nas v tem primeru zanimata, sta "DataProcesed" in "ErrMsg". Če bo naš zahtevek uspešen in se vmes ne bodo pojavile napake, dobimo v znački "ErrMsg" odgovor true, niz v znački "ErrMsg" pa bo ostal prazen, v nasprotnem primeru bo v

znački "ErrMsg" pisalo false, v spremenljivki napaka pa bo opis napake, zakaj klic spletne storitve ni bil uspešen.

```
<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Body>
    <ProcessData>
      <ProcessDataResponse>
        <DataProcesed>true</DataProcesed>
        <errMsg>napaka</errMsg>
      </ProcessDataResponse>
    </ProcessData>
  </soap:Body>
</soap:Envelope>
```

Slika 10: Primer odgovora v XML obliki. Ker je v znački DataProcesed zapisano true, pomeni, da bo spremenljivka "napaka" prazna

## 6.6 Tipi datotek

B2B vmesnik ni omejen na sprejem in oddajo točno določenih tipov datotek. Posredujemo mu lahko datoteke s kakršnokoli končnico in B2B vmesnik ne bo imel težav s pošiljanjem in prejemanjem. V večini primerov pa bomo sicer uporabili datoteke s končnicama .txt in .xml, lahko pa sta ti dve vrsti datotek tudi arhivirani v datotekah s končnicama, na primer .zip in .rar.













### 6.6.1 Txt in xml datoteke

Če na B2B sprejmemo oziroma pošljemo običajno datoteko z eno izmed zgoraj omenjenih končnic, jo enostavno posredujemo dalje. Na primer, če od partnerja dobimo datoteko s končnico .txt, jo zapišemo v bazo in v arhiv brez dodatnih operacij.

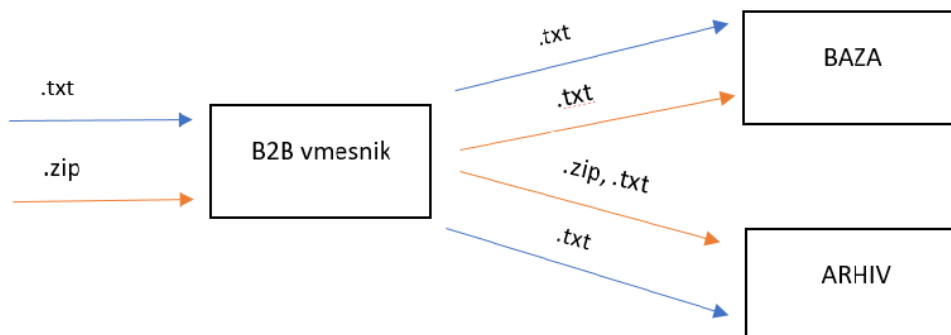
### 6.6.2 Zip in rar datoteke

V kolikor na B2B dobimo arhivirano datoteko, jo obravnavamo drugače kot datoteko, ki ni arhivirana. Nearhivirano datoteko bomo zapisali v bazo in arhiv, arhivirano pa najprej razpakiramo, nato pa vsebujoče datoteke zapišemo

v bazo, arhivirano datoteko pa skupaj z vsebujočimi datotekami shranimo v arhiv.

	630_P_ABC.D20170419.T050112.txt	24.4.2017 7:40	Text Document	72 KB
	631_S_ABC.D20170419.T050002.txt	24.4.2017 7:40	Text Document	16 KB
	632_ABCRAZVOJ_PROMET_D20170419.T0...	24.4.2017 7:40	Compressed (zipp...	5 KB
	633_P_ABC.D20170419.T050112.txt	24.4.2017 13:28	Text Document	72 KB
	634_S_ABC.D20170419.T050002.txt	24.4.2017 13:28	Text Document	16 KB
	635_ABCRAZVOJ_PROMET_D20170419.T0...	24.4.2017 13:28	Compressed (zipp...	5 KB
	636_P_ABC.D20170413.T050059.txt	25.4.2017 8:16	Text Document	91 KB
	637_S_ABC.D20170413.T050006.txt	25.4.2017 8:16	Text Document	17 KB
	638_ABCRAZVOJ_PROMET_D20170413.T0...	25.4.2017 8:16	RAR File	6 KB
	639_P_ABC.D20170413.T050059.txt	25.4.2017 8:17	Text Document	91 KB
	640_S_ABC.D20170413.T050006.txt	25.4.2017 8:17	Text Document	17 KB
	641_ABCRAZVOJ_PROMET_D20170413.T0...	25.4.2017 8:17	Compressed (zipp...	6 KB

Slika 11: Arhivirane in nearhivirane datoteke v arhivu



Slika 12: Prikaz shranjevanja .zip in .txt datotek. V arhiv se shrani tako .zip datoteka, kot vsebovana .txt, v bazo pa se shrani zgolj .txt datoteka

## 6.7 Nastavitve

V bazi imamo pripravljeno tabelo, iz katere preberemo podatke, ki jih bomo uporabili za prenos podatkov. Vsak partner ima v tabeli svoj vnos, v katerem so shranjene nastavitve, kaj se zgodi z njegovimi datotekami, ko jih sprejmemo. V tej tabeli so shranjeni tudi podatki za shranjevanje datotek



v arhivski direktorij. Za partnerje bomo v tabeli shranjevali podatke o tem, kakšen tip prenosa datotek si želimo. Imeli bomo tri tipe prenosa:

- L - datoteko bomo iz B2B vmesnika prenesli v bazo in na datotečni arhiv
- M - datoteko bomo prenesli na direktorij, ki je podan v stolpcu "dir prenosa" ter na arhiv
- V - metoda združuje obe prejšnji, torej datoteko prenese na direktorij iz "dir prenosa", v bazo ter v arhiv.

V primeru, da je kot partner podan »B2B«, bomo parameter1 uporabili kot arhivski direktorij za oddane datoteke, parameter2 pa kot arhiv za sprejete datoteke, parametra tip prenosa in direktorij prenosa v tem primeru ne bosta v uporabi. Če bi se pojavila potreba po dodatnih nastavitvah, ki bi jih potrebovali, imamo na voljo še stolpca parameter4 in parameter5. Primer zapisov za posameznega partnerja, na primer "JURED00":

- APLIKACIJA - JUR000,
- PARAMETER1 - neuporabljen parameter (uporabljen samo, če je aplikacija B2B),
- PARAMETER2 - neuporabljen parameter (uporabljen samo, če je aplikacija B2B),
- PARAMETER3 - neuporabljen parameter (uporabljen samo, če je aplikacija B2B),
- PARAMETER4 - zaenkrat neuporabljen parameter,
- PARAMETER5 - zaenkrat neuporabljen parameter,
- TIP PRENOSA - V,
- DIR PRENOSA - /B2B/TEST/TEST - direktorij, kamor program premakne prispelo datoteko

APLIKACIJA	PARAMETER1	PARAMETER2	PARAMETER3	PARAMETER4	PARAMETERS	TIP_PRENOSA	DIR_PRENOSA
UFPDpovratnica						L	
DBTEST						V	/B2B/ARHIV/TEST
TEST2						M	/B2B/ARHIV/TEST
TEST3						L	
TEST5						V	/B2B/ARHIV/TEST
TEST7						V	/B2B/ARHIV/TEST
B2B	/B2B/ARHIV/ODDAJA	/B2B/ARHIV/SPREJEM	/B2B				
JUREDOO						V	/B2B/ARHIV/TEST
UPINKZ						V	/B2B/ARHIV/TEST
B2B_RAZV	/B2B/ARHIV/ODDAJA	/B2B/ARHIV/SPREJEM	/B2B				

Slika 13: Nastavitve v bazi

## 6.8 Branje in pisanje datotek v/iz baze

### 6.8.1 SQL prepared Statement

Za branje in pisanje iz baze uporabljamo Oraclov vmesnik PreparedStatement, ki nam olajša sestavljanje SQL ukazov. Z uporabo PreparedStatementa je izvajanje SQL ukazov bolj priročno. Objekt PreparedStatement izhaja iz bolj splošnega razreda Statement.

Če želimo izvesti Statement večkrat, takrat uporaba PreparedStatementa običajno skrajša čas izvedbe ukaza.

Glavna prednost preparedStatementa je, da je v večini primerov SQL pozvedba poslana takoj v bazo, kjer se izvede. Objekt PreparedStatement ne vsebuje samo SQL stavka, ampak tudi stavek SQL, ki je bil že prej preveden. To pomeni, da ko se PreparedStatement izvrši, lahko baza izvrši poizvedbo PreparedStatement, ne da bi jo še enkrat prevajala.

Čeprav se objekti PreparedStatement lahko uporabljajo za stavke SQL brez parametrov, se jih najpogosteje uporablja za stavke, ki sprejmejo dodatne parametre. Prednost uporabe stavkov SQL s parametri je, da lahko uporabimo isti stavek večkrat in pri tem le zamenjamo parametre. [3]

### 6.8.2 Branje datotek iz baze

Pri branju datotek iz baze se prek Oracle WebLogic Serverja najprej povežemo na bazo. Nato s pomočjo Oracle PreparedStatementa preberemo željeno datoteko. Ko smo prebrali datoteko, jo zapišemo na željeno lokacijo (v našem

primeru na "outbox" direktorij), kjer bo na voljo B2B vmesniku, da jo pošlje partnerjem. Datoteki se bo pri prenosu atribut smer spremenil na "O" (out).

### 6.8.3 Pisanje datotek v bazo

V obratnem postopku datoteke, ki smo jih prejeli na B2B, zapišemo v bazo in jim nastavimo enolično identifikacijsko številko, ime datoteke ter datum vpisa. Datoteko prevzamemo iz "inboxa", ji nastavimo parametre in jo zapišemo v bazo. Datoteki se bo pri prenosu atribut smer nastavil na "I" (in).

## 6.9 Pisanje logov

Pri prenosih datotek iz baze in v bazo kličemo tudi proceduro, ki v tabelo zapisuje vse spremembe ter prenose na B2B vmesniku. Log tabela:

LOG_TKST	LOG_VJR	LOG_CLOB	CREATED_ON	CREATED_BY	UPDATED_ON	UPDATED_BY
2016 Operacija: preberi_eno...	B2B servlet	(HugeClob)	13.4.2017 7:38:09	SKUP		
2015 Operacija: preberi_eno...	B2B servlet	(HugeClob)	13.4.2017 7:38:09	SKUP		
2014 Operacija: preberi_eno...	B2B servlet	(HugeClob)	13.4.2017 7:38:08	SKUP		
2013 Operacija: preberi_eno...	B2B servlet	(HugeClob)	13.4.2017 7:36:18	SKUP		
2012 Operacija: preberi_eno...	B2B servlet	(HugeClob)	13.4.2017 7:36:17	SKUP		
2011 Operacija: preberi_eno...	B2B servlet	(HugeClob)	13.4.2017 7:36:16	SKUP		
2010 Operacija: preberi_eno...	B2B servlet	(HugeClob)	13.4.2017 7:29:22	SKUP		
2009 Operacija: preberi_eno...	B2B servlet	(HugeClob)	13.4.2017 7:29:22	SKUP		
2008 Operacija: preberi_eno...	B2B servlet	(HugeClob)	13.4.2017 7:29:22	SKUP		
2007 Operacija: preberi_eno...	B2B servlet	(HugeClob)	13.4.2017 7:29:22	SKUP		
2006 Operacija: preberi_eno...	B2B servlet	(HugeClob)	13.4.2017 7:29:20	SKUP		
2005 Operacija: preberi_eno...	B2B servlet	(HugeClob)	13.4.2017 7:29:20	SKUP		
2004 Operacija: preberi_eno...	B2B servlet	(HugeClob)	13.4.2017 7:29:20	SKUP		
2003 Operacija: preberi_eno...	B2B servlet	(HugeClob)	12.4.2017 15:44:00	SKUP		
2002 Operacija: preberi_eno...	B2B servlet	(HugeClob)	12.4.2017 15:44:00	SKUP		
2001 Operacija: poslji...	B2B servlet	(HugeClob)	12.4.2017 15:37:48	SKUP		
2000 Operacija: poslji...	B2B servlet	(HugeClob)	12.4.2017 15:37:48	SKUP		
1999 Operacija: poslji...	B2B servlet	(HugeClob)	12.4.2017 15:37:48	SKUP		
1998 Operacija: preberi_eno...	B2B servlet	(HugeClob)	12.4.2017 15:13:58	SKUP		
1997 Operacija: preberi_eno...	B2B servlet	(HugeClob)	12.4.2017 15:13:58	SKUP		
1996 Operacija: preberi...	B2B servlet	(HugeClob)	12.4.2017 15:11:55	SKUP		
1995 Operacija: preberi...	B2B servlet	(HugeClob)	12.4.2017 15:11:55	SKUP		
1994 Operacija: preberi...	B2B servlet	(HugeClob)	12.4.2017 15:11:55	SKUP		
1993 Operacija: preberi...	B2B servlet	(HugeClob)	12.4.2017 15:11:55	SKUP		
1992 Operacija: preberi...	B2B servlet	(HugeClob)	12.4.2017 15:11:55	SKUP		
1991 Operacija: poslji...	B2B servlet	(HugeClob)	12.4.2017 15:06:32	SKUP		
1990 Operacija: poslji...	B2B servlet	(HugeClob)	12.4.2017 15:06:32	SKUP		
1989 Operacija: poslji...	B2B servlet	(HugeClob)	12.4.2017 15:06:32	SKUP		

Slika 14: Seznam logov v bazi

Pri uspešnem prenosu datoteke v log zapišemo vrsto operacije, ID partnerja, tip prenosa, datum in uro prenosa in ime prenesene datoteke. Nov zapis v log se naredi za vsak prenos posebej (prenos v bazo, prenos v arhiv). Zapis v primeru uspešnega prenosa:

---

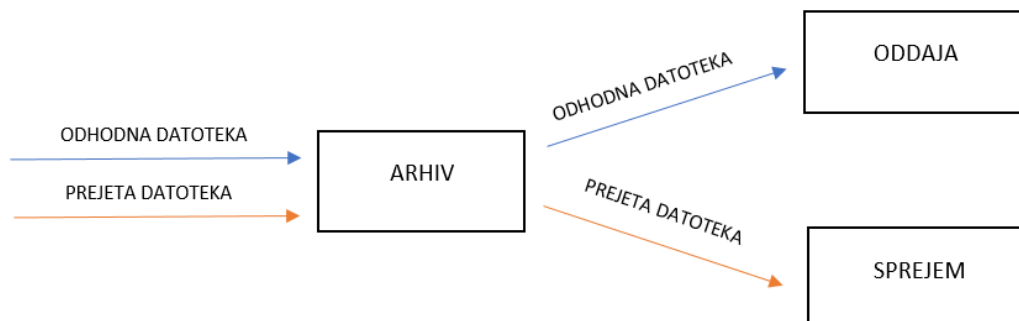
```
Operacija: preberi_eno  
ID partnerja: TEST5  
Tip prenosa: V  
Datum in ura: 21/04/2017 07:49:51  
Ime prenešene datoteke: TEST5_20141027.XML  
Prenešeno v arhiv: . B2B/ARHIV/SPREJEM
```

Slika 15: Primer loga

V primeru neuspešnega prenosa, torej če podamo napačno pot, napačen id ali napačno arhivsko pot, pa v bazo zapišemo vrsto operacije, datum in uro in opis napake.

## 6.10 Shranjevanje v arhiv

Vse prenešene datoteke bomo tudi prenesli tudi v datotečni arhiv. Tako so datoteke lažje dostopne vsem uporabnikom, ki želijo do njih dostopati. V tabeli nastavitev preberemo pot za shranjevanje dohodnih in odhodnih datotek v arhiv. Vse odhodne datoteke se bodo shranjevale v direktorij »oddaja«, vse sprejete pa v »sprejem«. Datoteke bodo ob tem pred imenom dobile tudi zaporedno številko vnosa.



Slika 16: Struktura arhiva

69_DBSRAZVOJ_DBSTEST_20141027.XML	30.3.2017 9:23	XML File	7.115 KB
70_DBSRAZVOJ_DBSTEST_20141027.XML	30.3.2017 9:23	XML File	7.115 KB
71_UPPD_DBSPOD_NAK201700003.xml.gpg	30.3.2017 9:38	GPG File	2 KB
72_UPPD_DBSPOD_GOT201700009.xml.gpg	30.3.2017 11:27	GPG File	2 KB
73_UPPD_DBSPOD_NAK201700004.xml.gpg	30.3.2017 13:28	GPG File	2 KB
74_UPPD_DBSPOD_GOT201700010.xml.gpg	30.3.2017 13:29	GPG File	2 KB
75_UPPD_DBSPOD_NAK201700005.xml.gpg	30.3.2017 13:30	GPG File	2 KB
76_RZDIUZ_AJPES_20170111.XML	30.3.2017 13:48	XML File	9.109 KB
77_RZDIUZ_AJPES_20141010	31.3.2017 8:41	File	3.761 KB
78_RZDIUZ_AJPES_20141010	31.3.2017 8:41	File	3.761 KB
79_RZDIUZ_AJPES_20141010	31.3.2017 8:42	File	3.761 KB
80_DBSRAZVOJ_DBSTEST_20141027.XML	31.3.2017 8:42	XML File	7.115 KB
81_RZDIUZ_AJPES_20141028.XML	31.3.2017 8:42	XML File	7.250 KB
82_RZDIUZ_AJPES_20141010	31.3.2017 8:47	File	3.761 KB
83_DBSRAZVOJ_DBSTEST_20141027.XML	31.3.2017 8:47	XML File	7.115 KB
84_RZDIUZ_AJPES_20141028.XML	31.3.2017 8:47	XML File	7.250 KB
85_RZDIUZ_AJPES_20141024.XML	31.3.2017 8:54	XML File	7.048 KB
86_RZDIUZ_AJPES_20141010	31.3.2017 9:28	File	3.761 KB
87_TEST_310317.XML	31.3.2017 9:44	XML File	9.109 KB
88_RZDIUZ_AJPES_20141010	31.3.2017 10:03	File	3.761 KB

Slika 17: Seznam datotek v arhivu

## 6.11 Načini klicanja

### 6.11.1 B2B

Klic WS se opravi avtomatsko pri sprejemu datoteke na B2B vmesnik. Klic opravi B2B vmesnik, ki SOAP XML ovojnico pošlje spletni storitvi v obdelavo. Pri tem klicu WS uporabi podatke iz tabele nastavitve in datoteko prenese na želene lokacije, torej v bazo ali arhiv. V SOAP XML ovojnici B2B WSju poda podatke o imenu partnerja, poti do datoteke ter dodatne parametre. WS iz teh podatkov sestavi ukaze in izvede primerne klice. WS

ustvari primeren XML odgovor in ga pošlje nazaj na B2B. Če med prenosom ne pride do napake, je prenos uspešen, v nasprotnem primeru pa se v značko ErrorMessage zapiše napaka, ki je tudi prikazana na zaslon v grafičnem delu B2B vmesnika.

### **6.11.2 Klic iz baze**

Klic WS je možen tudi neposredno iz baze oz. iz orodja Toad for Oracle. Pri klicu se povežemo na spletno storitev, sestavimo SOAP XML sporočilo ter ga pošljemo na WS, ki izvede ustrezne operacije in kreira odgovor ter nam ga dostavi.

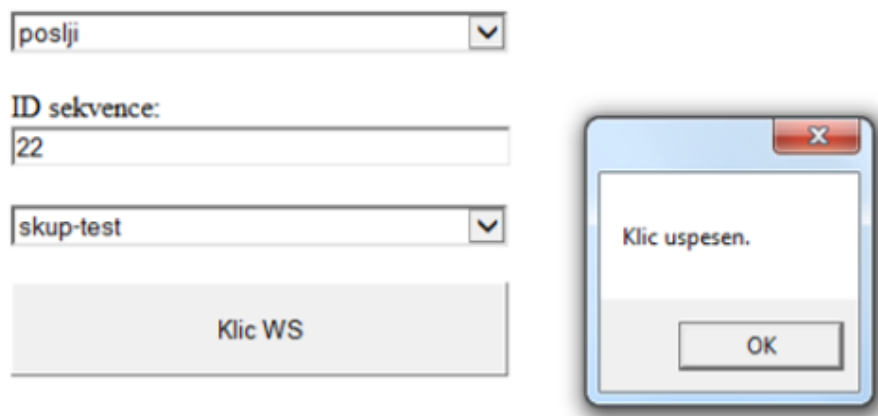
### **6.11.3 Klic iz posebne aplikacije**

Za uporabo metod »pošlji« in »preberi« lahko uporabimo tudi dodatno namizno aplikacijo, ki nam omogoča enostaven klic WS brez sestavljanja SOAP ovojníc in dodatnega pisanja programske kode. Grafični vmesnik temelji na Visual Basic Scriptu ter HTML značkah in CSS oblikovanju. Omogoča nam izbiro med operacijama »pošlji« in »preberi«. Ob izbiri metode nam vmesnik avtomatsko ponudi vnos samo tistih parametrov, ki jih dejansko potrebujemo. V primeru operacije »pošlji« nam bo vmesnik ponudil zgolj možnost vnosa parametrov »ID sekvence« in »Ime baze«, pri operaciji preberi pa lahko podamo pot do direktorija, ki ga želimo prebrati in ime baze. Grafični vmesnik nato avtomatsko generira SOAP XML ovojnico, sestavi URL naslov spletne storitve ter ovojnico pošlje na WS. V kolikor je bil klic uspešen, se pojavi pojavno okno z izpisom »klic uspešen«, v nasprotnem primeru pa se nam na zaslon izpiše napaka, poleg napake pa je zapisan tudi razlog, ki je preprečil uspešen klic WS.

Primer grafičnega vmesnika:

<div> <div>izberi metodo</div> <div> <div>izberi metodo</div> <div>poslji</div> <div>preberi</div> </div> </div>	<div>poslji</div> <div>ID sekvence:</div> <div></div> <div>skup-test</div> <div>Klic WS</div>
<div>Ime direktorija:</div> <div></div> <div>skup-test</div> <div>Id partnerja:</div> <div></div> <div>Package File Name:</div> <div></div> <div>Klic WS</div>	

Slika 18: Primer izbire metode na grafičnem vmesniku in zakritja atributov, ki jih ne potrebujemo



Slika 19: Primer uspešnega klica WS iz grafičnega vmesnika

## 6.12 Od lokalne do produkcijske verzije

Spletno storitev smo izdelali v treh verzijah in sicer v lokalni, testni in produkcijski.

### 6.12.1 Lokalna verzija

V procesu razvoja je bila spletna storitev najprej testirana v razvojnem okolju na lokalnem računalniku ter strežniku in lokalni bazi s testnimi podatki. V tej fazi spletna storitev še ni bila na voljo B2B vmesniku ali uporabnikom.

### 6.12.2 Testna verzija

V naslednji fazi je bila storitev naložena na testni Weblogic strežnik, kjer je bilo omogočeno tudi testiranje z zunanjimi testnimi B2B partnerji. Po zaključku testiranja z zunanjimi partnerji in znotraj banke je bila končna verzija spletne storitve prenešena na produkcijski strežnik, kar pomeni da



je bila prenešena na nov strežnik z novim naslovom, kjer bo na voljo vsem uporabnikom znotraj banke in zunanjim partnerjem.

### **6.12.3   Produksijska verzija**

Zaradi prilagodljivosti spletne storitve lahko produkcijsko verzijo brez težav nadgradimo in dodamo nove funkcionalnosti, v kolikor bi to zahteval kateri izmed uporabnikov oziroma bi se za to pojavila potreba znotraj banke.



## Poglavje 7

# Sklepne ugotovitve

V sklopu diplomskega dela je bil implementiran B2B vmesnik s podporo spletne storitve, ki banki olajšuje komunikacijo s strankami ter izmenjavo datotek. Spoznali smo delovanje grafičnega dela B2B vmesnika, spletnih storitev ter Oracleovih baz. V glavnem delu je opisan grafični del B2B vmesnika z vsemi pripadajočimi nastavitvami ter podporni del, torej spletna baza, spletni strežnik in spletna storitev. Vse te komponente povežejo celoten projekt v usklajeno celoto. B2B vmesnik tako močno pripomore k boljši komunikaciji banke s strankami, poleg tega pa olajša delo vsem zaposlenim v banki in skrajša proces izmenjave datotek. Natančneje smo se spoznali tudi z orodji Toad, Weblogic Server in JDeveloper, ki se uporabljajo za razvoj programske opreme s pomočjo Jave in SQL-a.



# Literatura

- [1] Douglas Bell and Parr Mike. *Java for Students*. Prentice Hall Europe, 1998.
- [2] Search Microservices. WeblLogic. <http://searchmicroservices.techtarget.com/definition/WebLogic/>, 2017. [Online; Dostopano 25.7.2017].
- [3] Oracle. preparedstatement. Dosegljivo: <http://docs.oracle.com/javase/tutorial/jdbc/basics/prepared.html>, 2017. [Online; Dostopano: 7. 3. 2017].
- [4] Avrom Roy-Faderman, Peter Koletzke, and Paul Dorsey. *Oracle JDeveloper 10g Handbook*. The McGraw-Hill Education, 2004.
- [5] Bert Scalzo and Dan Hotka. *Toad toad description*. Sams Publishing, 2003.
- [6] Wikipedia. Webservice. Dosegljivo: [https://en.wikipedia.org/wiki/Web\\_service](https://en.wikipedia.org/wiki/Web_service), 2017. [Online; Dostopano 1. 9. 2017].
- [7] Wikipedia. Xml. <https://en.wikipedia.org/wiki/XML>, 2017. [Online; Dostopano 31.8.2017].
- [8] Toad World. Toad for oracle. <https://www.toadworld.com/m/freeware/544>, 2017. [Online; Dostopano 25.7.2017].
- [9] ZBS. B2b. Dosegljivo: <http://www.zbs-giz.si/zdruzenje-bank.aspx?StructureId=1200>, 2017. [Online; Dostopano 27. 8. 2017].